Extracción automática de estructuras argumentativas en textos de opinión cubanos mediante proyección de etiquetas y aprendizaje profundo

Automatic extraction of argumentative structures in Cuban opinion texts through label projection and deep learning

Luis Ernesto Ibarra Vázquez,[1] **Damian Valdés Santiago**[1]
[1]Facultad de Matemática y Computación, Universidad de La Habana, Cuba
luise98cu@gmail.com, dvs89cs@matcom.uh.cu

Resumen: La Extracción de Argumentos se realiza tradicionalmente mediante anotación manual de expertos en lingüística, lo que demora mucho tiempo. Este artículo propone aplicar algoritmos de aprendizaje profundo al campo de la Extracción de Argumentos en textos de la prensa cubana, constituyendo el primero de su tipo publicado y adaptado para textos del español de Cuba, hasta donde los autores conocen. Para ello, 1) se crean conjuntos de datos a partir de provenientes del idioma inglés, 2) se proponen y entrenan los modelos y 3) se anotan automáticamente las unidades de discurso argumentativas (UDA). Los atributos utilizados para la representaciones de los textos son aprendidos en el proceso de entrenamiento para ajustarse al criterio argumentativo de los datos. De los conjuntos de datos disponibles, se realizó un análisis de las ventajas y deficiencias de cada uno para la anotación de las "Cartas a la Dirección" del periódico cubano *Granma*. Los resultados obtenidos en la extracción de UDAs alcanzaron valores de F1 = 0,82 comparados con 0,85 del estado del arte. En las demás tareas, los resultados no son directamente comparables con los del estado del arte, los mejores valores F1 obtenidos fueron 0,56 en la clasificación de UDAs, 0,74 en la predicción de enlaces y 0,39 en la clasificación de enlaces.
Palabras clave: Extracción de argumentos, procesamiento de lenguaje natural, aprendizaje profundo.

Abstract: Argument Extraction is traditionally performed by manual annotation by linguistic experts, which takes lots of time. This paper proposes deep learning algorithms to perform the Argument Extraction in Cuban press texts, constituting the first of its kind published and adapted for Cuban Spanish texts, as far as the authors knowledge. To this end, 1) datasets are created by annotation projection from other ones in English language, 2) models are proposed and trained, and 3) automatic annotation of Argumentative Discourse Units (ADUs) is performed. The features used for text representations are learned in the training process to match the argumentative criteria of the data. From the available data sets, an analysis of the advantages and deficiencies of each one was made for the annotation of the "Letters to the Editor" of the Cuban newspaper *Granma*. The results obtained in the extraction of ADUs reached values of F1 = 0.82 compared to 0.85 of the state of the art. In the other tasks, the results are not directly comparable with those of the state of the art, the best F1 values obtained were 0.56 in ADU classification, 0.74 in link prediction and 0.39 in link classification.
Keywords: Argument extraction, natural language processing, deep learning.

1 Introducción

La argumentación es una actividad verbal, social y racional destinada a convencer a un crítico razonable de la aceptabilidad de un punto de vista mediante la presentación de proposiciones que justifican o refutan la proposición expresada en el punto de vista (Van Eemeren y Grootendorst, 2004).

Varias tareas en el Procesamiento de Lenguaje Natural (PLN) se han desarrollado al-

rededor de diferentes problemas relacionados con la argumentación. Entre estas se encuentran: el minado de opiniones, sentimientos y emociones expresadas en un texto (Liu, 2010), la detección de controversias, y la zonificación argumentativa.

Es necesario realizar un análisis de los argumentos dados, para transformar el texto no estructurado a datos argumentativos que permitan el entendimiento de los puntos de vista y de cómo se "apoyan" o "atacan" entre sí. Este análisis es posible realizarlo manualmente o utilizando programas especializados para la anotación, aunque la práctica ha demostrado que este proceso requiere de una gran cantidad de tiempo y de personal calificado (Eger et al., 2018).

La Extracción de Argumentos (EA) es la rama del PLN encargada del estudio de métodos para la extracción automática de las estructuras argumentativas de los textos y su posterior procesamiento (Lawrence y Reed, 2020). Esta tarea se divide en cuatro subtareas fundamentales: i) la extracción y ii) clasificación de las componentes argumentativas del texto, y iii) la extracción y iv) clasificación de las relaciones entre estas.

La EA se caracteriza por la poca disponibilidad de datos anotados y por la heterogeneidad de las anotaciones. Además, la gran mayoría de los estudios realizados en el campo se encuentran en idiomas como el inglés, alemán o chino (Eger et al., 2018). En español, se reportan pocas investigaciones del análisis de los argumentos (Esteve, Casacuberta, y Rosso, 2020) y, en Cuba, no se encontró ninguna referencia, según la búsqueda de literatura científica realizada por los autores.

El objetivo, de esta investigación es proponer un algoritmo basado en aprendizaje profundo para la extracción y análisis de estructuras argumentativas en textos de la prensa cubana (en particular, la sección "Cartas a la Dirección" del periódico *Granma*), constituyendo el primero de su tipo publicado y adaptado para textos del español de Cuba, hasta donde los autores conocen. Para lograr dicho objetivo, en primer lugar, es necesario obtener mediante *crawling* los textos a analizar del sitio web del periódico *Granma*. Luego, se proponen algoritmos de aprendizaje automático capaces de realizar las tareas de EA sobre estos textos, que requiren conjuntos de datos anotados en español sobre los cuales se puedan entrenar.

Para la extracción de argumentos se presentan dos modelos, el primero se encarga de la segmentación y clasificación de las componentes argumentativas mediante la clasificación de los *tokens* en etiquetas BIOES, que delimitan y clasifican las unidades de discurso argumentativas (UDA). En el segundo, se analizan las posibles relaciones entre UDAs de manera independiente para saber si están relacionadas o no y el tipo de relación existente. Los modelos utilizan redes neuronales convolucionales (CNN, en inglés), *Long Short Term Memory* (LSTM, en inglés) (Hochreiter y Schmidhuber, 1997) y *Conditional Random Field* (CRF, en inglés) (Lafferty, McCallum, y Pereira, 2001) como elementos principales en sus arquitecturas, además se emplean vectores GloVe (Pennington, Socher, y Manning, 2014) para la representación de las palabras.

El artículo se divide en varias secciones. Primero, se presentan las definiciones relativas a la argumentación y la EA. Luego, se presenta un estado del arte de la EA con una discusión de las ventajas y desventajas de cada enfoque y se introduce la proyección de corpus. Más adelante, se presentan los modelos propuestos para resolver el problema en cuestión. A continuación se muestran los resultados del entrenamiento de los modelos y en la anotación de los textos de "Cartas a la Dirección". Finalmente, se exponen las conclusiones y recomendaciones de la investigación.

2 Extracción de Argumentos

La EA consiste en la identificación y extracción automática de las estructuras de inferencia y razonamiento expresadas como argumentos presentes en el lenguaje natural (Lawrence y Reed, 2020). La EA permite dar respuesta a este problema presentando los argumentos y cómo sus relaciones justifican las posiciones del hablante. Dicho problema está constituido por diferentes estructuras y se compone de distintas tareas necesarias para su solución.

Existen diferentes estudios que conforman una metodología de análisis para identificar los argumentos. El modelo de Toulmin (Toulmin, 2003) introduce categorías con distintas funciones dentro de la argumentación. En el idioma español existen rasgos lingüísticos que, además de dar indicación de la existencia de argumentos, dan pie para conocer las

relaciones entre estos y los tipos de argumentos. Venegas (2005) determina 16 categorías y 51 rasgos lingüísticos, dando una idea de la gran variedad de marcadores presentes en la argumentación.

2.1 Estructuras Argumentativas

Las estructuras argumentativas son las partes de la argumentación de los textos y sus relaciones. Estas se componen de dos elementos principales: las Unidades de Discurso Argumentativas (UDAs) y los enlaces o relaciones existentes entre estas. Las UDAs corresponden a la unidad mínima de argumentación, definida como un segmento de texto que juega un solo rol para el argumento analizado, y es delimitado por segmentos vecinos que tienen roles diferentes o ningún rol (Stede y Schneider, 2018).

Las UDAs se relacionan entre sí conformando el proceso de inferencia y razonamiento del argumento. Tanto los enlaces como las UDAs son clasificados en dependencia de su rol en la argumentación. Estas clasificaciones parten de los conceptos de afirmación, declaración controversial y parte central del argumento, y premisa, razones que la justifican o refutan, y en las relaciones de ataque y apoyo.

2.2 Tareas de extracción de argumentos

Dada la definición de estructuras argumentativas y que el objetivo de la EA es extraerlas, se conciben las siguientes tareas principales:

1. Extracción de UDAs: separar los segmentos de texto que formarán parte de la UDA.
2. Clasificación de UDAs: asignar una categoría argumentativa a la UDA segmentada.
3. Extracción de relaciones entre las UDAs: determinar si están relacionadas las UDAs o no.
4. Clasificación de relaciones entre las UDAs: asignar una categoría a la relación extraída.

2.3 Variantes para la Extracción de Argumentos

Varias investigaciones han dado respuesta a los problemas asociados a EA, mostrando una variedad en enfoques y métodos. Para la segmentación de las UDAs se ha separado en oraciones y luego clasificado cada una en si es UDA o no mediante algoritmos como *Naive Bayes* (NB) y máquinas de soporte vectorial (SVM, en inglés) (Palau y Moens, 2009; Goudas et al., 2015). Otras aproximaciones para esta tarea consiste en la clasificación en etiquetas BIO de los *tokens* del texto (Goudas et al., 2015; Stab y Gurevych, 2017; Eger, Daxenberger, y Gurevych, 2017) y en el uso de reglas basadas en anotaciones lingüísticas (Dykes et al., 2020).

En las tareas de predicción y clasificación de enlaces se han empleado gramáticas libre de contexto basadas en anotaciones de los *tokens* (Palau y Moens, 2009). SVM y aprendizaje profundo han sido utilizados para clasificar las posibles relaciones dos a dos (Goudas et al., 2015; Galassi, 2021), en Goudas et al. (2015) se optimiza la estructura final con un problema de optimización lineal en enteros.

Las UDAs y las relaciones han sido representadas de diferentes maneras, ya sea por atributos escogidos a mano mediante conocimiento experto (Palau y Moens, 2009; Goudas et al., 2015), como por atributos aprendidos por los algoritmos en la fase de entrenamiento (Eger, Daxenberger, y Gurevych, 2017; Galassi, 2021).

En los modelos propuestos (ver secciones 3.1 y 4.1), gran parte de las representaciones son aprendidas en el proceso de entrenamiento y las que se agregan de forma manual casi no influyen en la escalabilidad del sistema. Cuando se trata de unir los resultados de los dos modelos, hay una propagación de errores, aunque se utiliza el modelado de problemas conjuntos para minimizarlo.

2.4 Proyección de etiquetas

La proyección de etiquetas es un algoritmo donde se transfieren las etiquetas de un corpus anotado a nivel de *tokens* en un lenguaje origen hacia su traducción en un lenguaje objetivo. En (Eger et al., 2018) se propone un algoritmo de proyección a partir de las alineaciones de palabras. El proceso se divide en varias partes:

1. Traducción automática de oraciones: proceso de traducir automáticamente texto de un lenguaje fuente a un lenguaje objetivo.
2. Alineación de palabras: consiste en asignar las palabras del lenguaje fuente a sus

equivalentes generadas en el lenguaje objetivo.

3. Proyección de etiquetas: consiste en transformar las etiquetas de las palabras en la secuencia origen hacia las palabras de la secuencia destino tomando como datos las alineaciones entre estas.

3 Segmentación y clasificación de UDAs

Las tareas de segmentación y clasificación de UDAs se resuelven conjuntamente. Para esto se modela como un problema secuencia a secuencia cuyo objetivo es asignar, a los *tokens* extraídos del documento entrada, una etiqueta BIOES para segmentar las UDAs. Para la clasificación del tipo de UDA, al conjunto de etiquetas BIES se le añade otra etiqueta que representa el tipo de UDA. Con este esquema se obtiene una cantidad de etiquetas $|\{B, I, E, S\}| \cdot |Clasificaciones\,de\,UDA| + |\{O\}|$.

3.1 Modelo de segmentación y clasificación de UDAs

Sea D un documento entrada, este es separado en una secuencia de n *tokens* D_i, donde n es la mayor longitud encontrada en los documentos del conjunto de datos (si la cantidad de *tokens* es menor que n entonces D_i es completado con un *token* especial de enmascarado). A cada *token* se le asigna su representación vectorial GloVe de dimensión $g = 300$, dando como resultado $G_{ij} \in \mathbb{R}^{n \times g}$. Esta representación inicial presenta información semántica de las palabras y conserva las relaciones espaciales entre ellas.

Para la representación de información morfológica de la palabra se construyen dos codificadores que procesan los caracteres de cada *token* y devuelven una representación vectorial de estos. A cada caracter se le asigna un vector que será entrenado convirtiendo un *token* en un vector de dimensión $q \times c$, donde q es el tamaño máximo de palabra en el conjunto de datos y c es la dimensión del vector asignado a cada caracter.

Uno de los modelos entrenados está basado en CNN, este modelo entrena una representación de caracteres de dimensión $cd = 50$, representando un *token* como un vector de dimensión $q \times cd$. Se conforma por una capa de convolución unidimensional con $f = 30$ filtros y un kernel de tamaño $k = 3$, seguida por una capa *max pooling* que convierte la secuencia en un vector de dimensión $1 \times f$, que luego es concatenado a la representación del *token* a que pertenece.

Otro modelo utilizado para calcular una representación morfológica está basado en RNN. Se usó un modelo LSTM bidireccional con dimensión $l = 25$ para calcular la representación del *token*, para las dimensiones de los caracteres se utilizaron vectores de tamaño l, el resultado final constituye la concatenación de la corrida hacia adelante y hacia atrás, formando una representación de dimensión $1 \times 2 \cdot l$ del *token*. Este vector es concatenado a la representación del *token* correspondiente.

Otro atributo usado en la representación de los *tokens* constituyen las etiquetas de partes de la oración de estos. El conjunto de etiquetas elegido es un conjunto universal (Petrov, Das, y McDonald, 2012) aplicable a muchos idiomas. Estas etiquetas se representan como un vector al que se le asigna 1 en la posición correspondiente a la clase y 0 en los otros elementos (codificación *one-hot*) y este es transformado por una capa densa con $p = 5$ neuronas y función de activación *ReLU*. El resultado se concatena a la representación del *token* correspondiente. Mediante la extracción de estos atributos el *token* es representado en tres maneras: semántica, morfológica y estructural, con el objetivo de que sean aprendidos los rasgos lingüísticos correspondientes.

Del proceso de vectorización se obtiene un vector con dimensión $n \times t$, donde t es la dimensión final de la representación de los *tokens* Este vector es modificado por una capa LSTM bidireccional de dimensión $m = 200$. A esta salida se le añade una conexión residual al ajustarle la dimensión con una capa densa. Luego, la secuencia es procesada por una capa densa de dimensión $k = 100$ con activación *ReLU*, produciendo una representación final de dimensión $n \times k$. Finalmente, se utiliza una capa CRF para la clasificación final de la secuencia en las etiquetas finales. El resultado final constituye un vector de dimensión n que representa las clasificaciones inferidas por el modelo (Figura 1).

Para prevenir el sobreajuste se agregaron capas de normalización y de *dropout* (0.5) entre cada proceso y se usaron regularizaciones L2 y *dropout* en las capas densas y LSTM. Para prevenir el sobreentrenamiento se aplicó

una terminación temprana cuando no se encontró una mejora de la función de pérdida en el conjunto de validación por más de 10 épocas consecutivas. Como optimizador se utilizó Adam con una tasa de aprendizaje de 0,001.

La salida del modelo es procesada para eliminar los errores en las etiquetas BIOES, errores como segmentos que no empiecen en B o terminen en E, o segmentos con más de una clasificación, obteniendo así un formato BIOES válido.

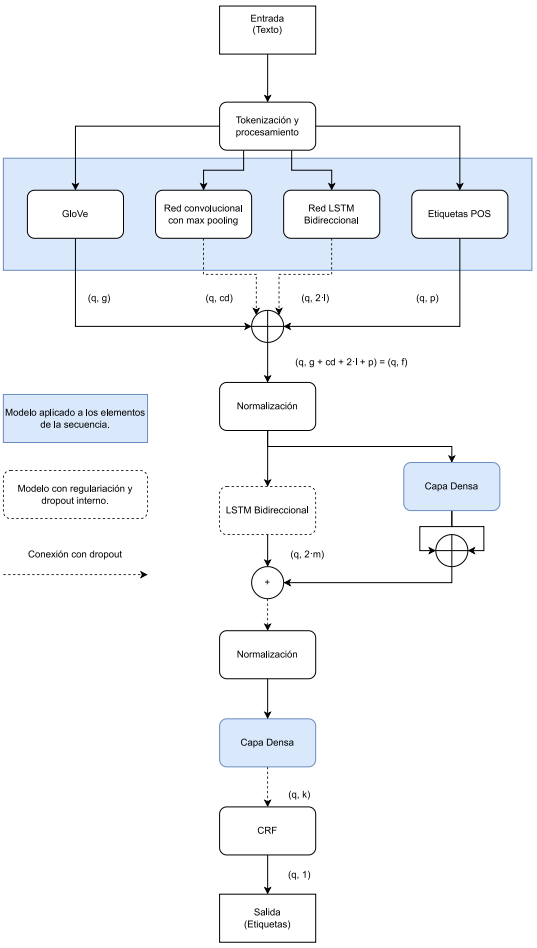

Figura 1: Segmentador de UDAs.

4 Predicción y clasificación de enlaces

Las tareas de extracción y clasificación de enlaces son modeladas de forma conjunta. El problema consiste en clasificar pares de UDAs, representando origen y objetivo del enlace, en el tipo de relación que existen entre estas. Como tarea auxiliar se clasifican los tipos de UDAs que intervienen en la relación. La salida del modelo constituye en una tupla de tres elementos: la clasificación de la relación, la clasificación de la UDA origen, la clasificación de la UDA objetivo. Si el enlace existe o no, es calculado a partir del vector de probabilidades obtenido de la clasificación de la relación.

4.1 Modelo de predicción y clasificación de enlaces

Sean dos UDAs, S y T, donde S representa la fuente de la relación, mientras que T representa al objetivo. Estas secuencias son tokenizadas y se les asigna la representación GloVe de cada palabra, obteniendo dos vectores de dimensión $u \times g$, donde u es el tamaño máximo de UDAs en el conjunto de entrenamiento y $g = 300$ es la dimensión del *embedding*.

Estos vectores son modificados por una red densa compuesta por $ca = 4$ capas con activación *ReLu* de dimensiones 50, 50, 50 y 300 respectivamente, añadiendo una conexión residual a la salida de esta. El próximo paso consiste en aplicar una capa densa de dimensión $di = 50$ y luego un *average pooling* de tamaño $dp = 10$, obteniendo vectores de dimensión $\frac{q}{dp} \times di$. Estos vectores son modificados por un LSTM bidireccional con $lm = 50$ unidades.

La salida de los procesamientos es concatenada con la distancia argumentativa, obteniendo una representación conjunta de la relación a analizar. Esta representación es modificada por una red residual obteniendo una representación final de dimensión $l = 20$ y luego sometida a los clasificadores de relación y de tipos de UDAs (Figuras 2 y 3).

Para prevenir el sobreajuste se agregaron capas de normalización y de *dropout* entre cada proceso y se usaron regularizaciones L2 y *dropout* en las capas densas y LSTM, todos los *dropout* tienen valor $dr = 0,1$. Para prevenir el sobreentrenamiento se aplicó una terminación temprana cuando no se encontró una mejora de la función de pérdida en el conjunto de validación durante $v = 5$ épocas consecutivas. Como optimizador se utilizó el algoritmo de Adam con descenso exponencial y tasa de aprendizaje $lr = 0{,}003$.

Dado que se realiza un aprendizaje de varias tareas, se tienen varias funciones de pérdida individuales que conforman la función de pérdida final e. Sea e_r la función de pérdida de la clasificación de la relación, e_s la del tipo de UDA origen y e_t del tipo de UDA objetivo, entonces $e = 10 \cdot e_r + e_s + e_t$ (Galassi, 2021).

4.2 Preprocesamiento de predicción y clasificación de enlaces

Las UDAs extraídas son agrupadas de dos en dos y anotadas con su distancia argumentativa, solo seleccionando los pares que no se enlacen con ellos mismos y que su distancia sea menor que 10 (Para disminuir el número de pares a analizar). Al conjunto de entrenamiento se añade las representaciones inversas de las relaciones, por ejemplo, si $a \xrightarrow{c} b$ entonces se agregara el par $b \xrightarrow{c^{-1}} a$, donde c^{-1} es una nueva clasificación de relación que representa el inverso de la clasificación c. Este proceso se realiza para aumentar la cantidad de relaciones positivas en el conjunto entrenante.

4.3 Posprocesamiento de predicción y clasificación de enlaces

A partir de la distribución de probabilidades de las relaciones devueltas por el modelo, se calcula si el par está enlazado o no. Para esto, las categrías vinculadas a las clases de relaciones originales se suman, y si superan el 50 %, se considera enlazado el par.

5 Conjuntos de Datos

Para el entrenamiento de los modelos propuestos se utilizaron corpus diferentes, estos presentan esquemas de anotación distintos entre sí, difiriendo principalmente en la definición de UDA y las clasificaciones dadas a estas y a las relaciones.

Todos los conjuntos de datos están originalmente en inglés, por lo tanto, se les aplicó el algoritmo de proyección de corpus para obtener uno en español para ser usado en el entrenamiento de los modelos.

Para la traducción automática se utilizó el servicio de Google Translate, obteniendo las alineaciones de palabras con AwesomeAlign (Dou y Neubig, 2021). Con estos datos se realizó la proyección de etiquetas con el algoritmo propuesto por Eger et al. (2018).

5.1 Corpus Ensayos Argumentativos

Este corpus (Stab y Gurevych, 2017) presenta 402 documentos, divididos por los autores en 286 documentos para entrenamiento (70 %), 80 para prueba (20 %) y 36 para validación (10 %). El corpus contiene ensayos

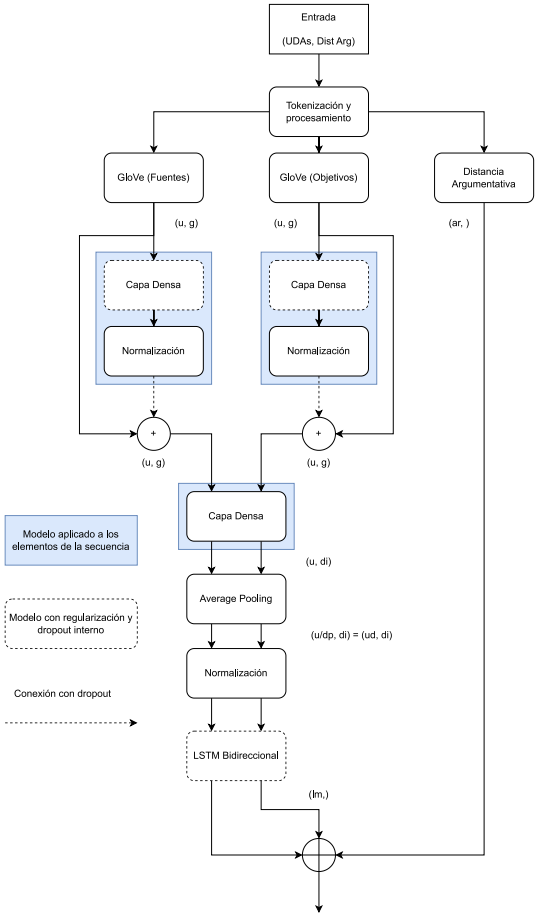

Figura 2: Predictor de enlaces.

de estudiantes en los que se argumentan sobre temas como cooperar o competir y sobre las contribuciones de la tecnología a la sociedad. Las anotaciones de las UDAs se conforman por segmentos de textos argumentativos, clasificados en *MajorClaim* con 751 (12 %), *Claim* con 1506 (25 %) y *Premise* con 3832 (63 %).

La estructura de las relaciones entre los UDAs conforman árboles en los que las *Major Claim* del texto son las raices. Solo se permiten relaciones entre *Premise-Premise* y *Premise-Claim*, clasificados en ataque con 219 (6 %) y apoyo con 3613 (93 %). Las relaciones entre *Claim* y *MajorClaim* se indican de manera diferente, con una calificación de la *Claim* de si está a favor (1228) o en contra (278) de las *MajorClaim* del documento. Estas anotaciones se convirtieron en relaciones de *attack* y *support* respectivamente, lo que resultó en un número final de 715 (10 %) de ataque y 5958 (90 %) de soporte.

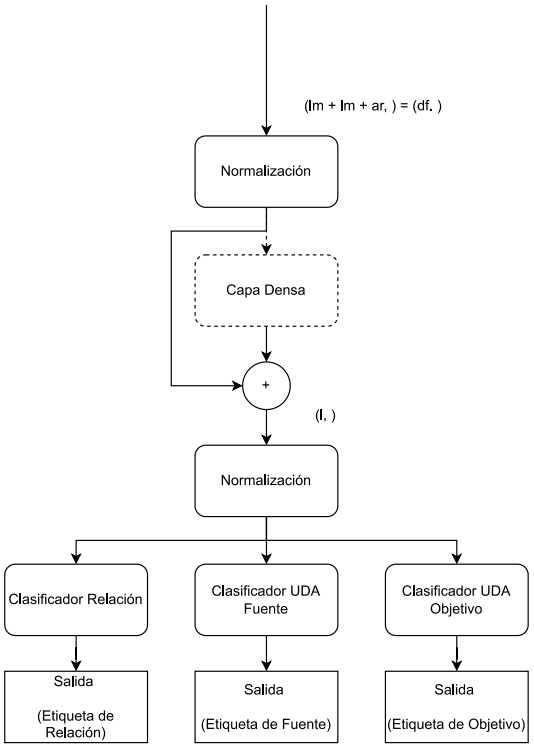

Figura 3: Predictor de enlaces (continuación).

5.2 CDCP

El corpus CDCP (Niculae, Park, y Cardie, 2017) está conformado por 731 comentarios de usuarios extraídos de la web sobre el tema de prácticas de cobro de deudas a los consumidores. Las UDAs se encuentran segmentadas en oraciones y todas se consideran argumentativas. Están clasificadas en *policy* con 815 (17 %), *value* con 2180 (44 %), *fact* con 785 (16 %), *testimony* con 1116 (22 %) y *reference* con 32 (1 %). Las relaciones se encuentran clasificadas en *reason* con 1352 (95 %) y *evidence* con 73 (5 %).

5.3 AbsTRCT

El corpus AbsTRCT (Mayer, Cabrio, y Villata, 2020) se compone de 500 documentos sobre el estudio de cuatro enfermedades diferentes: glaucoma, hipertensión, hepatitis B y diabetes. Cada oración es una UDA, aunque no todas son consideradas argumentativas. Estas se clasifican en *MajorClaim* con 93 (3 %), *Claim* con 993 (30 %) y *Premise* con 2198 (67 %). Las relaciones están representadas por tres categorías: *support* con 1763 (85 %), *partial-attack* con 238 (12 %) y *attack* con 60 (3 %).

Estos conjuntos de datos son pequeños para los modelos a entrenar, además, presentan desbalance en las clases existentes.

5.4 Cartas a la Dirección

La sección "Cartas a la Dirección" (Gallego Ramos y Rosabal García, 2013) es un segmento del periódico *Granma* donde se publican cartas enviadas por la población o empresas a dicha entidad. En general, las cartas presentan dudas o problemas de la población con el objetivo de obtener respuestas del organismo asociado.

Mediante *crawling*, se extrajeron 2891 cartas desde el 30 de agosto del 2013 hasta el 28 de octubre del 2022. Estas contienen aproximadamente 975000 palabras en los datos y, en promedio, la cantidad de palabras por carta es 330. Se encontraron 874 cartas en respuesta a cartas enviadas, lo que representa un 30 % del total. De las cartas, se seleccionaron las que fueran en respuesta a otra y también las cartas que fueron respondidas para tener una mayor concentración de cartas que fueran argumentativas, esta selección está conformada por 1702 cartas, lo que representa un 59 % del total de cartas.

6 Resultados

Para realizar la selección del modelo se utilizó el corpus de Ensayos Argumentativos. Con este se ajustaron las arquitecturas e hiperparámetros de los modelos propuestos. La mejor combinación de estos fue utilizada para el entrenamiento de los corpus restantes. Finalmente, los modelos fueron utilizados para anotar los textos de la sección "Cartas a la Dirección" de *Granma*.

6.1 Segmentador de UDA

El modelo seleccionado fue usado en el entrenamiento de los demás conjuntos de datos obteniendo los resultados mostrados en Tabla 1 y Tabla 2.

Las métricas 50 %F1 y 100 %F1 (Persing y Ng, 2016) están basadas en la idea de la métrica F1, pero orientada a secuencias, donde el número denota el porcentaje de secuencia inferida que debe coincidir con la secuencia anotada para ser considerado una coincidencia.

En las tablas se observa una diferencia entre los valores de F1 Ponderado y de Macro F1, dadas por el pobre balance de las clases que hace que las menos representadas sean más difíciles de ser correctamente anotadas. Los valores mayores de 50 %F1 en compara-

Corpus	Ensayos Argumentativos	CDCP	AbsTRCT
F1 Ponderado	0,76	0,65	0,86
Macro F1	0,56	0,45	0,50
Accuracy	0,77	0,66	0,87
100 %F1	0,72	0,61	0,61
50 %F1	0,83	0,68	0,75

Tabla 1: Métricas de las pruebas del segmentador de UDA.

Corpus	Ensayos Argumentativos	CDCP	AbsTRCT
F1 Ponderado	0,89	0,95	0,90
Macro F1	0,82	0,56	0,79
Accuracy	0,89	0,96	0,91
100 %F1	0,81	0,82	0,66
50 %F1	0,94	0,93	0,82

Tabla 2: Métricas BIOES de las pruebas del segmentador de UDA.

ción con 100 %F1 indican que el modelo logra inferir las posiciones de las UDA de manera general, pero sus límites se hacen más complejos de discernir.

6.2 Predictor de Enlaces

Para el modelo se realizó un voto conjunto del ensamblado de tres modelos, dado que el entrenamiento está basado en la aleatoriedad, se entrenan los modelos con los mismos datos obteniendo inferencias no necesariamente iguales.

En el entrenamiento del modelo en los demás conjuntos de datos se obtuvieron los resultados de las Tablas 3 y 4.

Corpus	Macro F1	*Accuracy*
Ensayos argumentativos	0,33	0,57
CDCP	0,37	0,63
AbsTRCT	0,39	0,61

Tabla 3: Métricas de clasificación de relaciones de las pruebas del predictor de enlace.

En la Tabla 3 se observan valores más discretos que en la Tabla 4 en ambas métricas. Esta diferencia en la métrica Macro F1 se interpreta como el fallo del modelo en predecir correctamente la clase de la relación. En la tarea de predicción de enlace el modelo se desempeña mejor, aunque con diferencias entre los conjuntos de datos, dando a entender que la estructura de las relaciones de estos pueden influir en el resultado.

Corpus	Macro F1	*Accuracy*
Ensayos argumentativos	0,68	0,75
CDCP	0,79	0,68
AbsTRCT	0,83	0,74

Tabla 4: Métricas de predicción de relaciones de las pruebas del predictor de enlace.

7 Evaluación cualitativa de la EA

Dado que las estructuras argumentativas varían en su forma en cada corpus es complejo realizar un método que evalúe de forma justa los resultados obtenidos por los diferentes modelos de manera conjunta. Una variante sería anotar las cartas con los esquemas argumentativos presentes en los conjuntos de datos, esto constituye una labor en la que se requiere personal experto, previo estudio y preparación, además de tiempo.

Por ello, el proceso que se llevó a cabo en esta investigación para realizar la validación consistió en un análisis cualitativo realizado a criterio del autor. Para esto se seleccionaron 15 pares de cartas, la carta original y la respuesta enviada a esta. Cada una de estas 30 cartas fueron anotadas por los modelos entrenados en cada conjunto de datos y se realizó una evaluación que consideró si la UDA se extrajo y clasificó correctamente, así como si la relación también fue extraída y clasificada por el modelo de manera adecuada.

7.1 Análisis del corpus Ensayos Argumentativos

Los ensayos argumentativos presentan una anotación de UDAs a un nivel de unidades de texto que pueden ser más pequeñas que oraciones y clasifican estas en las clases *MajorClaim* (MC), *Claim* (C) y *Premise* (P). Las relaciones se clasifican en de *supports* y *attacks*.

En general, se observan problemas en la segmentación de UDAs debido al formato y dominio del texto. Las cartas presentan una estrutura donde al final se realiza una firma poniendo información acerca del remitente. Esta estructura no contribuye a la argumentación, pero el modelo en varias ocasiones detecta componentes en estas. Otro problema se observa en la extracción de supuestas UDAs sin componente argumentativo, generalmente, estos elementos, si se expanden, pueden

lograr establecer una mejor UDA.

Ejemplos donde el modelo propuesto no fue exitoso:

- [en cada uno de los establecimientos de nuestra Cadena de Tiendas]$_{MC}$: incompleto, mejora incorporando elementos de la izquierda (No a todos los productos con próxima fecha de vencimiento se le aplica rebaja de precios).
- [Esperamos lo antes posible una solución]$_P$: en contexto, no contiene información que lo haga premisa.

Ejemplos donde el modelo fue exitoso:

- pudiese [contribuir al ahorro de agua y la prestación de un mejor servicio]$_C$
- [es que estamos limitados de este servicio, y no desde hace un tiempo, es que nunca lo hemos tenido]$_P$

Las relaciones anotadas por el modelo tienden a contener falsos positivos, además dado que este conjunto de datos posee un gran desbalance en las etiquetas de las relaciones favoreciendo estas a las de *supports*, el modelo no fue capaz de realizar anotaciones de *attacks*, tanto en el conjunto de pruebas como en las "Cartas a la Dirección" del *Granma*.

7.2 Análisis de CDCP

El corpus CDCP las UDAs son segmentadas, en la mayoría de los casos, en oraciones (solamente el 1 % de los *tokens* se encuentran fuera de una UDA), estas son clasificadas en *testimony* (T), *fact* (F), *policy* (P), *reference* (R) y *value* (V). Las relaciones presentan dos tipos de relaciones *evidences* y *reasons*.

Los errores más comunes cometidos por el modelo propuesto en la segmentación, provienen del uso de signos de puntuación que no representan un cambio de oración, en estos casos se separan las UDAs. También existen errores de clasificación incorrecta, de, por ejemplo, *testimony* que podrían ser *fact*.

Ejemplos deonde el modelo propuesto no fue exitoso:

- [Junto a la misiva se le entregó al Inass certificados de salarios devengados y las tarjetas sn2-25.]$_T$: se clasifica mejor como *fact*.
- [Caridad Real Gutiérrez, Jefe de Trámites y Pensiones, Inass.]$_T$: firma de la carta como elemento argumentaivo.

Ejemplos donde el modelo propuesto fue exitoso:

- [Mi jubilación comenzó el 29 de febrero de 2016, no el 29 de febrero de 2017.]$_T$
- [No se sabe cuánto queda, lo que obliga al cliente a estar haciendo cuentas constantemente.]$_F$

La cantidad de relaciones anotadas por el modelo entrenado en este corpus disminuye en comparación a las anotadas por el modelo entrenado con el corpus Ensayos Argumentativos. Las relaciones *reasons* son las más encontradas.

7.3 Análisis del corpus AbsTRCT

El conjunto de datos presenta un estilo de segmentación de UDAs en donde se anotan secciones de textos más grandes que en el corpus Ensayos Argumentativos, aunque no necesariamente todas las oraciones o la oración completa es considerada argumentativa. Estas se clasifican igual que el corpus Ensayos Argumentativos, aunque en este conjunto de datos se presenta un desbalance de etiquetas grande, favoreciendo a las *Premise* y las *Claim*, dejando sin representación casi a *MajorClaim* (menor del 1 % de las etiquetas BIOES), lo que trajo como consecuencia que el modelo no fuera capaz de diferenciar este tipo de UDA. Las relaciones se presentaron como *partial-attack*, *attack* y *support*, influenciadas también por la poca cantidad de relaciones de *attack*.

En la clasificación de UDAs se evidencia una gran cantidad de *Premise*.

Ejemplos deonde el modelo propuesto no fue exitoso:

- [, Director División Grandes Centros TRD Caribe.]$_C$: mala clasificación con mala segmentación y detección de *claim* en pie de firma de la carta.

Ejemplos donde el modelo propuesto fue exitoso:

- [Esta respuesta considera sin razón la preocupación de un lector, ¿así debe terminar la inquietud de un ciudadano, que confía en las instituciones con que cuenta la sociedad para enfrentar sus problemas?]$_C$

La cantidad de relaciones anotadas por el modelo entrenado en este corpus es la menor

de los demás conjuntos de datos. Se observa una gran cantidad de relaciones *support*. Las relaciones clasificadas como *partial-attack*, a consideración del autor, presentaron una baja precisión.

8 Discusión

Las comparaciones con el estado del arte se realizan por cada conjunto de datos y se muestran las métricas indicadas por los autores de cada propuesta. Cada corpus y propuesta presenta características únicas que hacen que difícil la comparación.

Una de las principales dificultades está dada por el hecho de que las métricas calculadas son de la versión proyectada al español, lo cual contribuye a variaciones en las etiquetas finales debido al lenguaje mismo o a errores en el proceso. Otros ejemplos en la dificultad de comparar las métricas se encuentra en los enfoques tomados por las investigaciones anteriores a la hora de realizar las tareas. En algunos casos la segmentación se presenta como una tarea de clasificación BIO, o se separan por oraciones y las clasifican en argumentativas o no.

En el aspecto de clasificación de las UDAs se emplean métodos como su clasificación independiente luego de ser extraída o su modelación conjunta con la segmentación. En la extracción y clasificación de relaciones se observan técnicas de optimización de problemas enteros, clasificación por SVM o también probando los posibles enlaces dos a dos independientemente.

En la comparación de métodos se seleccionaron seis métricas que evalúan las diferentes tareas de la EA. La métrica BIOES F1 se refiere a la Macro F1 de la clasificación de las etiquetas BIOES, esta constituye una medida que califica la tarea de segmentación de UDAs en el texto.

La métrica Clas UDA F1 es calculada como la Macro F1 de las etiquetas BIOES junto con las etiquetas del tipo de UDA, medida que evalúa la tarea de clasificación de las UDAs.

Rel Pred F1 es la medida Macro F1 de la predicción de enlaces y Rel Clas F1 la de la clasificación, estas son calculadas tomando en cuenta todos los pares seleccionados para el conjunto de datos.

En las Tablas 5-7 el símbolo ✓ significa que los algoritmos son directamente comparables, el símbolo $*$ expresa que el método de comparación es el mismo, pero no son usados los mismos elementos para calcular la métrica, y el símbolo × denota que la métrica no se computó en las investigaciones donde se propusieron los modelos.

Modelo	**BIOES F1**	**Clas UDA F1**	**Rel Pred F1**	**Rel Clas F1**
Propuesto	0,82	0,56	0,68	0,33
Stab y Gurevych (2017)	0,85 ✓	0,82	0,58	0,70
Niculae, Park, y Cardie (2017)	×	0,77	0,60	×
Galassi (2021)	×	0,53	0,36 *	0,18 *

Tabla 5: Métricas comparativas del corpus Ensayos Persuasivos.

Modelo	**BIOES F1**	**Clas UDA F1**	**Rel Pred F1**	**Rel Clas F1**
Propuesto	0,56	0,45	0,68	0,37
(Niculae, Park, y Cardie, 2017)	×	0,73	0,27	×
(Galassi, 2021)	×	0,79	0,30 *	0,15 *

Tabla 6: Métricas comparativas del corpus CDCP.

Modelo	**BIOES F1**	**Clas UDA F1**	**Rel Pred F1**	**Rel Clas F1**
Propuesto	0,79	0,50	0,74	0,39
(Mayer, Cabrio, y Villata, 2020)	×	0,88 ✓	×	0,66 *
(Galassi, 2021)	×	0,91	0,54 *	0,70 *

Tabla 7: Métricas comparativas del corpus AbsTRCT.

Se considera que el corpus CDCP se ajusta mejor a las características de las "Cartas a la Dirección". Este presenta orígenes similares y un conjunto de etiquetas de UDAs que se ajustan más a lo observado en las Cartas. También las Cartas presentan un alto contenido argumentativo, por lo que marcar todas las oraciones como argumentativas no constituye una fuente grande de errores.

Una desventaja de este esquema sobre otros es la carencia de una clasificación de las relaciones que implique un ataque, aunque esto se cubre con el hecho de que en los conjuntos en donde existen estas, los resultados son pobres en ese aspecto. La cantidad y calidad de relaciones, aunque tiene espacio para mejorar, es aceptable dada la dificultad del problema en EA.

La ventaja del modelo entrenado con el corpus de Ensayos Argumentativos en la extracción y clasificación de UDA es que utiliza

un conjunto de etiquetas que podría considerarse universal en la argumentación y además reduce el espacio de búsqueda de oraciones a segmentos de palabras, aunque estos puedan estar sujetos a errores.

La versión del modelo propuesto entrenado sobre el corpus AbsTRCT constituye el modelo con menor rendimiento. La clasificación de UDAs presentó una gran desproporción hacia *Premise* dejando muchas *Claim* sin ser correctamente clasificadas. Sobre las relaciones, reportó un nivel muy bajo de relaciones por documento, respecto a las que se podrían formar.

9 Conclusiones

En la investigación se logró la extracción de estructuras argumentativas en los textos de las "Cartas a la Dirección" del periódico *Granma*. Para esto se hizo un análisis de los modelos entrenado con los distintos conjuntos de datos y se seleccionó el modelo que más se ajustaba al dominio de las cartas. Esta selección se realizó sin tener un conjunto anotado por lingüistas de las Cartas, por lo que los autores fueron los que establecieron los criterios cualitativos para la selección del modelo final.

En los resultados obtenidos en las tareas de segmentación y clasificación de UDAs se observan valores 50 %F1 entre 0,82 y 0,94 y 0,68 y 0,83, respectivamente, indicando una segmentación aceptable pero que en ocasiones falla a la hora de clasificar correctamente. Al predecir los enlaces y clasificarlos los modelos obtienen resultados de Macro F1 entre 0,68 y 0,83 y 0,33 y 0,39, respectivamente. Estos evidencian una mayor dificultad a la hora de trabajar con las relaciones, sobre todo al clasificarlas. Las comparaciones con las investigaciones previas con los resultados de los modelos entrenados se vieron dificultadas por los diferentes enfoques presentados en estas a la hora de seleccionar cómo modelar el problema y cómo procesar los datos para entrenar los modelos.

Este trabajo aportó nuevos conjuntos de datos, estos son las "Cartas a la Dirección" extraídas del *Granma*, los corpus proyectados al español de Ensayos Argumentativos, AbsTRCT y CDCP y las Cartas anotadas con las estructuras argumentativas del modelo entrenado con el conjunto de datos CDCP. También presentó unos modelos capaces de adaptarse a los diferentes esquemas que se puedan presentar en la argumentación, haciéndolos viables para un estudio directo y sin el agrego de conocimiento específico de los datos.

El software implementado y los datos pueden encontrarse en `https://github.com/luisoibarra/argument-mining`.

10 Recomendaciones

La principal dificultad en el trabajo fue la carencia de un conjunto de anotados sobre el tema en específico relacionado con la extracción de argumentos en la prensa. Por lo que se propone la creación de un corpus anotado por lingüistas para poder realizar una mejor validación y entrenamiento del modelo propuesto. También se considera la creación de un servicio online basado en Brat[1] para la socialización y mejora de los resultados obtenidos. El uso de representaciones BERT (Devlin et al., 2019) ha llevado a mejorar los resultados de tareas del PLN (Mayer, Cabrio, y Villata, 2020), por lo tanto, se propone investigar el uso de estos *embeddings* en el modelo. El problema principal obtenido en el modelo fue relacionado con la predicción de enlaces, un problema que tiene el modelo es la falta de contexto global del texto para hacer la predicción, por lo que se insta a la búsqueda y experimentación de métodos que tomen esto en cuenta, una variante podrían ser las *Graph Neural Networks* (Wu et al., 2021).

Agradecimientos

Los autores agradecen el apoyo del Proyecto de Investigación "Dinámicas sociales, políticas y económicas en el discurso público en Cuba de principio del siglo XXI: estudios de CORESPUC", asociado al Programa Nacional de Ciencia y Técnica "Las Ciencias Sociales y las Humanidades. Desafíos ante la estrategia de desarrollo de la sociedad cubana", Código PN223LH011-011, Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), Cuba, 2021-2023.

Bibliografía

Devlin, J., M.-W. Chang, K. Lee, y K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

[1] `https://brat.nlplab.org/`

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), páginas 4171–4186, Minneapolis, Minnesota, Junio. Association for Computational Linguistics.

Dou, Z.-Y. y G. Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. En *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.

Dykes, N., S. Evert, M. Göttlinger, P. Heinrich, y L. Schröder. 2020. Reconstructing arguments from noisy text. *Datenbank-Spektrum*, 20(2):123–129, jul.

Eger, S., J. Daxenberger, y I. Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. En *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Eger, S., J. Daxenberger, C. Stab, y I. Gurevych. 2018. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! En *Proceedings of the 27th International Conference on Computational Linguistics*, páginas 831–844.

Esteve, M., F. Casacuberta, y P. Rosso. 2020. Minería de argumentación en el referéndum del 1 de octubre de 2017. *Procesamiento del Lenguaje Natural*, 65:59–66.

Galassi, A. 2021. *Deep Networks and Knowledge: from Rule Learning to Neural-Symbolic Argument Mining*. Ph.D. tesis, Alma Mater Studiorum - Università di Bologna.

Gallego Ramos, J. R. y A. Rosabal García. 2013. Las cartas sobre la mesa: Un estudio sobre la relación entre agenda pública y mediática en cuba: caso granma. *Signo y Pensamiento*, 32(62):98–113.

Goudas, T., C. Louizos, G. Petasis, y V. Karkaletsis. 2015. Argument extraction from news, blogs, and the social web. *International Journal on Artificial Intelligence Tools*, 24(05):1540024, oct.

Hochreiter, S. y J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lafferty, J., A. McCallum, y F. C. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. En *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, páginas 282–289.

Lawrence, J. y C. Reed. 2020. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818, 01.

Liu, B. 2010. Sentiment analysis and subjectivity. En *Handbook of Natural Language Processing*. Chapman and Hall/CRC, feb, páginas 651–690.

Mayer, T., E. Cabrio, y S. Villata. 2020. Transformer-based argument mining for healthcare applications. En *ECAI 2020-24th European Conference on Artificial Intelligence*.

Niculae, V., J. Park, y C. Cardie. 2017. Argument mining with structured SVMs and RNNs. En *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Palau, R. M. y M.-F. Moens. 2009. Argumentation mining. En *Proceedings of the 12th International Conference on Artificial Intelligence and Law*. ACM, jun.

Pennington, J., R. Socher, y C. D. Manning. 2014. Glove: Global vectors for word representation. En *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, páginas 1532–1543.

Persing, I. y V. Ng. 2016. End-to-end argumentation mining in student essays. En *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 1384–1394, San Diego, California, Junio. Association for Computational Linguistics.

Petrov, S., D. Das, y R. McDonald. 2012. A universal part-of-speech tagset. En *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, páginas 2089–2096.

Stab, C. y I. Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, sep.

Stede, M. y J. Schneider. 2018. Argumentation mining. *Synthesis Lectures on Human Language Technologies*, 11(2):1–191, dec.

Toulmin, S. E. 2003. *The Uses of Argument*. Cambridge University Press, jul.

Van Eemeren, F. H. y R. Grootendorst. 2004. *A systematic theory of argumentation: The pragma-dialectical approach*. Cambridge University Press.

Venegas, R. 2005. Hacia una identificación automatizada de rasgos argumentativos en corpus. *Discurso especializado e instituciones formadoras*, páginas 127–158.

Wu, Z., S. Pan, F. Chen, G. Long, C. Zhang, y P. S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, jan.